NORTH WESTERN UNIVERSITY, KHULNA



Course Title: Artificial Intelligence and Expert Systems Sessional

Course Code: CSE-3302

Lab Report

|  |  |
| --- | --- |
| **Submitted by:**  Name: Chandan Sourav Mallick  Id: 20201065010  Department of Computer Science and Engineering  North Western University, Khulna | **Submitted to:**  Tajul Islam  Departmental Head  Department of Computer Science and Engineering  North Western University, Khulna. |

**Submission Date:25.01.2023** **Teacher’s Signature**

**1. Algorithm Name: Breadth-First Search(BFS).**

**Description: The breadth-first search (BFS) algorithm is used to search a tree or graph data structure for a node that meets a set of criteria. It starts at the tree’s root or graph and searches/visits all nodes at the current depth level before moving on to the nodes at the next depth level. Breadth-first search can be used to solve many problems in graph theory.**

**Code Implementation using C-Programming:**

#include<stdio.h>

#include<conio.h>

/\*

Naming Convention-->

Name: Sourav Mallick

s = s\_forMatrixArray,

o = o\_forQuee,

u = u\_forVisitedArray,

r = r\_forNumberOfVartex,

a = a\_forLoop,

v = v\_forNastedLoop,

s = s\_forFront,

m = m\_forRare

\*/

int s\_forMatrixArray[20][20], o\_forQuee[20], u\_forVisitedArray[20], r\_forNumberOfVartex, a\_forLoop, v\_forNastedLoop, s\_forFront= 0, m\_forRare = -1;

void sourav\_forBFS(int vertex)

{

for(a\_forLoop = 1; a\_forLoop <= r\_forNumberOfVartex; a\_forLoop ++)

if(s\_forMatrixArray[vertex][a\_forLoop] && !u\_forVisitedArray[a\_forLoop])

o\_forQuee[++m\_forRare] = a\_forLoop;

if(s\_forFront <= m\_forRare)

{

u\_forVisitedArray[o\_forQuee[s\_forFront]] = 1;

sourav\_forBFS(o\_forQuee[s\_forFront++]);

}

}

void main()

{

printf("\t\t\t\t Hay Welcome...!");

printf("\n\t\t\t First Search(BFS)\n\n");

char name[200] = "Sourav Mallick";

printf("\nEnter the number of Vertex: ");

scanf("%d",&r\_forNumberOfVartex);

for(a\_forLoop=1; a\_forLoop <= r\_forNumberOfVartex; a\_forLoop++)

{

o\_forQuee[a\_forLoop] = 0;

u\_forVisitedArray[a\_forLoop] = 0;

}

printf("\nEnter graph data in matrix form:\n");

for(a\_forLoop=1; a\_forLoop<=r\_forNumberOfVartex; a\_forLoop++)

{

for(v\_forNastedLoop=1; v\_forNastedLoop<=r\_forNumberOfVartex; v\_forNastedLoop++)

{

scanf("%d",&s\_forMatrixArray[a\_forLoop][v\_forNastedLoop]);

}

}

int vertex;

printf("\nEnter the starting vertex:");

scanf("%d", &vertex);

sourav\_forBFS(vertex);

printf("\nThe node which are reachable are: \n");

for(a\_forLoop=1; a\_forLoop <= r\_forNumberOfVartex; a\_forLoop++)

{

if(u\_forVisitedArray[a\_forLoop])

printf("%d\t", a\_forLoop);

else

{

printf("\nBFS is not possible. Not all nodes are reachable....\n");

printf("Please Try Again...!!!");

break;

}

}

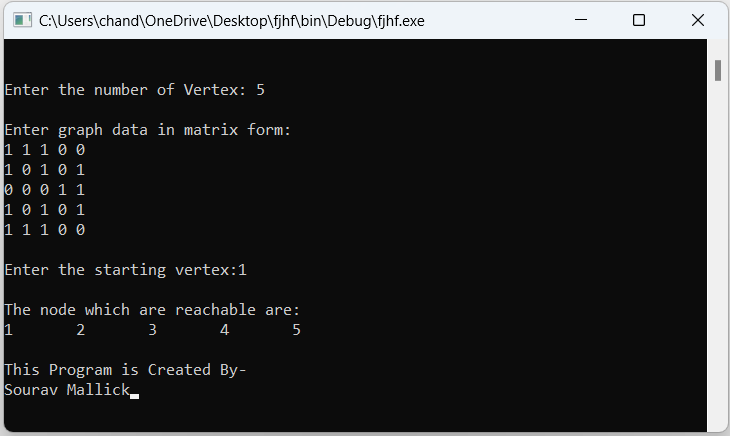
printf("\n\nThis Program is Created By- \n");

printf("%s",name);;

getch();

}

**Input & Output:**



**2. Algorithm Name: Depth-First Search(DFS).**

**Description: Depth-first search is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking.**

**So, the basic idea is to start from the root or any arbitrary node** **, mark the node, move to the adjacent unmarked node, and continue this loop until there is no unmarked adjacent node. Then backtrack and check for other unmarked nodes and traverse them. Finally, print the nodes in the path.**

**Code Implementation using C-Programming:**

#include<stdio.h>

#include<conio.h>

/\*

Naming Convention.

Sourav Mallick

s = s\_forAdjacencyMatrix,

o = o\_forQuee,

u = u\_forNumberOfVartex,

r = r\_forLoop,

a = a\_forNastedLoop,

v = v\_forCount,

\*/

int s\_forAdjacencyMatrix[20][20],o\_forQuee[20],u\_forNumberOfVartex,r\_forLoop , a\_forNastedLoop;

void sourav\_forDFS(int v)

{

o\_forQuee[v]=1;

for(r\_forLoop=1; r\_forLoop<=u\_forNumberOfVartex; r\_forLoop++)

if(s\_forAdjacencyMatrix[v][r\_forLoop] && !o\_forQuee[r\_forLoop])

{

printf("n %d->%d",v,r\_forLoop);

sourav\_forDFS(r\_forLoop);

}

}

void main()

{

int v\_forCount=0;

//clrscr();

printf("n Enter number of vertices:");

scanf("%d",&u\_forNumberOfVartex);

for(r\_forLoop=1; r\_forLoop<=u\_forNumberOfVartex; r\_forLoop++)

{

o\_forQuee[r\_forLoop]=0;

for(a\_forNastedLoop=1; a\_forNastedLoop<=u\_forNumberOfVartex; a\_forNastedLoop++)

s\_forAdjacencyMatrix[r\_forLoop][a\_forNastedLoop]=0;

}

printf("n Enter the adjacency matrix:n");

for(r\_forLoop=1; r\_forLoop<=u\_forNumberOfVartex; r\_forLoop++)

for(a\_forNastedLoop=1; a\_forNastedLoop<=u\_forNumberOfVartex; a\_forNastedLoop++)

scanf("%d",&s\_forAdjacencyMatrix[r\_forLoop][a\_forNastedLoop]);

sourav\_forDFS(1);

printf("n");

for(r\_forLoop=1; r\_forLoop<=u\_forNumberOfVartex; r\_forLoop++)

{

if(o\_forQuee[r\_forLoop])

v\_forCount++;

}

if(v\_forCount==u\_forNumberOfVartex)

printf("n Graph is connected");

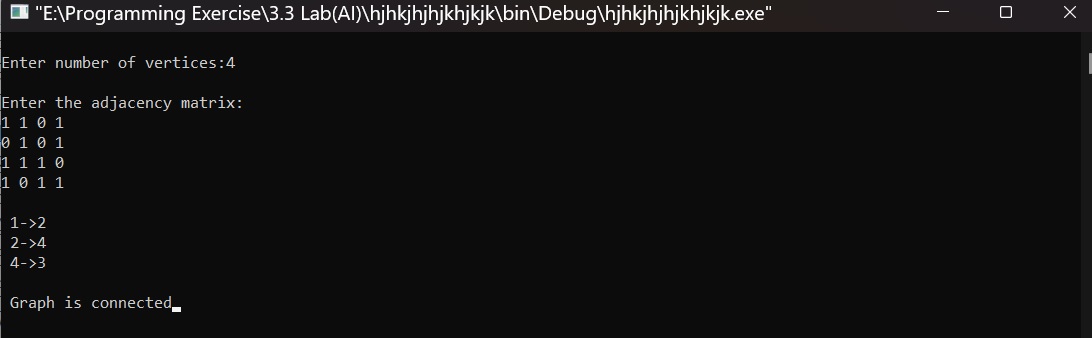
else

printf("n Graph is not connected");

getch();

}

**Input & Output:**

****

**3. Algorithm Name: Uniform-Cost Search(UCS).**

**Description: Uniform-cost search (UCS) is a search algorithm that works on search graphs whose edges do not have the same cost. In the previous examples, we did not mention or define any edge costs. In doing so, we treated every node as having the same cost. The cost of an edge can be interpreted as a value or loss that occurs when that edge is traversed. Mathematically, a cost is just a scalar value associated with some edge, and graphs with non-uniform cost edges tend to have a cost for every edge. Such a graph is known as a weighted graph.**

**Instead of exploring nodes in order of their depth from the root, like what BFS does, UCS expands nodes in order of their cost from the root. At each step, the next step n is chosen to be the one that minimizes a cost value g(n). g(n) is defined as the total cost of getting to a node n from the current position. The nodes are stored in a priority queue.**

**Code:**

#include <iostream>

#include <queue>

#include <vector>

using namespace std;

const int MAX = 100;

int cost[MAX][MAX], dist[MAX], parent[MAX];

bool visited[MAX];

int n;

struct Node {

int vertex;

int distance;

bool operator<(const Node& other) const {

return distance > other.distance;

}

};

void ucs(int start) {

for (int i = 0; i < n; i++) {

dist[i] = INT\_MAX;

visited[i] = false;

}

priority\_queue<Node> q;

dist[start] = 0;

q.push({start, 0});

while (!q.empty()) {

int u = q.top().vertex;

q.pop();

if (visited[u]) {

continue;

}

visited[u] = true;

for (int v = 0; v < n; v++) {

if (cost[u][v] != INT\_MAX && dist[u] + cost[u][v] < dist[v]) {

dist[v] = dist[u] + cost[u][v];

parent[v] = u;

q.push({v, dist[v]});

}

}

}

}

int main() {

cin >> n;

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

cin >> cost[i][j];

}

}

int start;

cin >> start;

ucs(start);

for (int i = 0; i < n; i++) {

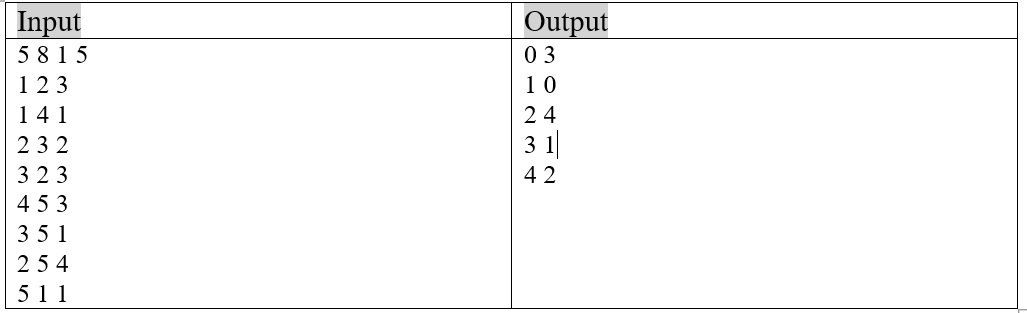
cout << i << " " << dist[i] << endl;

}

return 0;

}

**Input & Output:**

****

4.**Algorithm Name:** Genetic Algorithms(GAs).

Description: Genetic Algorithms(GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. **They are commonly used to generate high-quality solutions for optimization problems and search problems.**

**Genetic algorithms simulate the process of natural selection** which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate “survival of the fittest” among individual of consecutive generation for solving a problem. **Each generation consist of a population of individuals** and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits. This string is analogous to the Chromosome.

**Code:**

**#include <bits/stdc++.h>**

**using namespace std;**

**#define POPULATION\_SIZE 100**

**const string nam = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOP"\**

**"QRSTUVWXYZ 1234567890, .-;:\_!\"#%&/()=?@${[]}";**

**const string Target\_nam= "I am Sourav";**

**int random\_num(int start, int end)**

**{**

**int range = (end-start)+1;**

**int random\_int = start+(rand()%range);**

**return random\_int;**

**}**

**char mutated\_nam()**

**{**

**int len = nam.size();**

**int r = random\_num(0, len-1);**

**return nam[r];**

**}**

**string create\_gnome()**

**{**

**int len = Target\_nam.size();**

**string gnome = "";**

**for(int i = 0;i<len;i++)**

**gnome += mutated\_nam();**

**return gnome;**

**}**

**class Individual**

**{**

**public:**

**string chromosome;**

**int fitness;**

**Individual(string chromosome);**

**Individual mate(Individual parent2);**

**int cal\_fitness();**

**};**

**Individual::Individual(string chromosome)**

**{**

**this->chromosome = chromosome;**

**fitness = cal\_fitness();**

**};**

**Individual Individual::mate(Individual par2)**

**{**

**string child\_chromosome = "";**

**int len = chromosome.size();**

**for(int i = 0;i<len;i++)**

**{**

**float p = random\_num(0, 100)/100;**

**if(p < 0.45)**

**child\_chromosome += chromosome[i];**

**else if(p < 0.90)**

**child\_chromosome += par2.chromosome[i];**

**else**

**child\_chromosome += mutated\_nam();**

**}**

**return Individual(child\_chromosome);**

**};**

**int Individual::cal\_fitness()**

**{**

**int len = Target\_nam.size();**

**int fitness = 0;**

**for(int i = 0;i<len;i++)**

**{**

**if(chromosome[i] != Target\_nam[i])**

**fitness++;**

**}**

**return fitness;**

**};**

**bool operator<(const Individual &ind1, const Individual &ind2)**

**{**

**return ind1.fitness < ind2.fitness;**

**}**

**int main()**

**{**

**srand((unsigned)(time(0)));**

**int generation = 0;**

**vector<Individual> population;**

**bool found = false;**

**for(int i = 0;i<POPULATION\_SIZE;i++)**

**{**

**string gnome = create\_gnome();**

**population.push\_back(Individual(gnome));**

**}**

**while(! found)**

**{**

**sort(population.begin(), population.end());**

**if(population[0].fitness <= 0)**

**{**

**found = true;**

**break;**

**}**

**vector<Individual> new\_generation;**

**int s = (10\*POPULATION\_SIZE)/100;**

**for(int i = 0;i<s;i++)**

**new\_generation.push\_back(population[i]);**

**s = (90\*POPULATION\_SIZE)/100;**

**for(int i = 0;i<s;i++)**

**{**

**int len = population.size();**

**int r = random\_num(0, 50);**

**Individual parent1 = population[r];**

**r = random\_num(0, 50);**

**Individual parent2 = population[r];**

**Individual offspring = parent1.mate(parent2);**

**new\_generation.push\_back(offspring);**

**}**

**population = new\_generation;**

**cout<< "Generation: " << generation << "\t";**

**cout<< "String: "<< population[0].chromosome <<"\t";**

**cout<< "Fitness: "<< population[0].fitness << "\n";**

**generation++;**

**}**

**cout<< "Generation: " << generation << "\t";**

**cout<< "String: "<< population[0].chromosome <<"\t";**

**cout<< "Fitness: "<< population[0].fitness << "\n";**

**}**

**Input & Output:**

**Generation: 0 String: K j6K!MI6} Fitness: 10**

**Generation: 1 String: 5:ae615kQO[ Fitness: 10**

**Generation: 2 String: 5:ae615kQO[ Fitness: 10**

**Generation: 3 String: j{5\_m@oj6s] Fitness: 10**

**Generation: 4 String: KZGJ.fxwrE2 Fitness: 10**

**Generation: 5 String: j 5\_m@oj6s] Fitness: 9**

**Generation: 6 String: j 5\_m@oj6s] Fitness: 9**

**Generation: 7 String: j 5\_m@oj6s] Fitness: 9**

**Generation: 8 String: j 5\_m@oj6s] Fitness: 9**

**Generation: 9 String: j 5\_m@oj6s] Fitness: 9**

**Generation: 10 String: j 5\_m@oj6s] Fitness: 9**

**Generation: 11 String: j 5\_m@oj60] Fitness: 9**

**Generation: 12 String: j 5\_m@oj&s] Fitness: 9**

**Generation: 13 String: j 5\_m@oj6s] Fitness: 9**

**Generation: 14 String: j 5\_mjoj6s] Fitness: 9**

**Generation: 15 String: j 5\_Y@oj6s] Fitness: 9**

**Generation: 16 String: j 5\_mooj6s] Fitness: 9**

**Generation: 17 String: j 5\_m@oj6so Fitness: 9**

**Generation: 18 String: j 5\_m@oj6s] Fitness: 9**

**Generation: 19 String: j 5\_m@oj{s] Fitness: 9**

**Generation: 20 String: j 5\_(#oj(s] Fitness: 9**

**Generation: 21 String: j 5\_mjoj6s] Fitness: 9**

**Generation: 22 String: j 5\_m@oj2s] Fitness: 9**

**Generation: 23 String: j 5\_m@oj6s] Fitness: 9**

**Generation: 24 String: j 5\_m@ojus] Fitness: 9**

**Generation: 25 String: j 5\_m@ojus] Fitness: 9**

**Generation: 26 String: j 5\_0ooj6s] Fitness: 9**

**Generation: 27 String: j 5\_m#oj(s] Fitness: 9**

**Generation: 28 String: j 5\_0ooj6s] Fitness: 9**

**Generation: 29 String: j 5\_m#oj(s] Fitness: 9**

**Generation: 30 String: I 5\_m@ojaOL Fitness: 8**

**Generation: 31 String: I 5\_m@ojaOL Fitness: 8**

**Generation: 32 String: I 5\_m@ojaOL Fitness: 8**

**Generation: 33 String: I 5\_m@ojaOL Fitness: 8**

**Generation: 34 String: I 5mm@ojLOL Fitness: 7**

**Generation: 35 String: I 5mm@ojLOL Fitness: 7**

**Generation: 36 String: I 5mm@oJLOL Fitness: 7**

**Generation: 37 String: I 5mm@ojLOL Fitness: 7**

**Generation: 38 String: I 5mm@ojLOL Fitness: 7**

**Generation: 39 String: I U\_ @ojaOL Fitness: 7**

**Generation: 40 String: I 5mm@oJLOL Fitness: 7**

**Generation: 41 String: I 5mmGojLOL Fitness: 7**

**Generation: 42 String: I 5mm@ojLOL Fitness: 7**

**Generation: 43 String: I 5mmGojLOL Fitness: 7**

**Generation: 44 String: I 5mm@oJLOL Fitness: 7**

**Generation: 45 String: I 5mm@ojLOL Fitness: 7**

**Generation: 46 String: I 5mm@ojLOL Fitness: 7**

**Generation: 47 String: I 5mm@oJLOL Fitness: 7**

**Generation: 48 String: I 5mm@oJ8OL Fitness: 7**

**Generation: 49 String: I 5m @oJLOL Fitness: 6**

**Generation: 50 String: I 5m @oJLOL Fitness: 6**

**Generation: 51 String: I 5m @oJLOL Fitness: 6**

**Generation: 52 String: I 5m @oJLOL Fitness: 6**

**Generation: 53 String: I 5m @oJLOL Fitness: 6**

**Generation: 54 String: I 5m @oJLOL Fitness: 6**

**Generation: 55 String: I 5m @oJLOL Fitness: 6**

**Generation: 56 String: I 5m @oJLOL Fitness: 6**

**Generation: 57 String: I 5m @o.LOL Fitness: 6**

**Generation: 58 String: I 5m @o.LOL Fitness: 6**

**Generation: 59 String: I 5m @ou{-L Fitness: 5**

**Generation: 60 String: I 5m @ou{-L Fitness: 5**

**Generation: 61 String: I 5m @ou{-L Fitness: 5**

**Generation: 62 String: I 5m @ou{-L Fitness: 5**

**Generation: 63 String: I 5m @ou{-L Fitness: 5**

**Generation: 64 String: I 5m @ou{-L Fitness: 5**

**Generation: 65 String: I 5m @ou{-L Fitness: 5**

**Generation: 66 String: I 5m 8ou{-L Fitness: 5**

**Generation: 67 String: I 5m @ou{-L Fitness: 5**

**Generation: 68 String: I 5m @ou{-L Fitness: 5**

**Generation: 69 String: I 5m @ou{-L Fitness: 5**

**Generation: 70 String: I 5m @ou{aL Fitness: 4**

**Generation: 71 String: I 5m @ou{aL Fitness: 4**

**Generation: 72 String: I 5m @ou{aL Fitness: 4**

**Generation: 73 String: I 5m @ou{aL Fitness: 4**

**Generation: 74 String: I 5m @ou{aL Fitness: 4**

**Generation: 75 String: I 5m @ou{aL Fitness: 4**

**Generation: 76 String: I 5m @ou{aL Fitness: 4**

**Generation: 77 String: I 5m @ou{aL Fitness: 4**

**Generation: 78 String: I 5m @ou{aL Fitness: 4**

**Generation: 79 String: I 5m @ou{aL Fitness: 4**

**Generation: 80 String: I 5m @ou{aL Fitness: 4**

**Generation: 81 String: I 5m mou{aL Fitness: 4**

**Generation: 82 String: I 5m @ou{aL Fitness: 4**

**Generation: 83 String: I 5m @ou{aL Fitness: 4**

**Generation: 84 String: I 5m @ou{aL Fitness: 4**

**Generation: 85 String: I 5m :ou{aL Fitness: 4**

**Generation: 86 String: I 5m @ou{aL Fitness: 4**

**Generation: 87 String: I 5m mou{aL Fitness: 4**

**Generation: 88 String: I 5m @ou{aL Fitness: 4**

**Generation: 89 String: I 5m mou{aL Fitness: 4**

**Generation: 90 String: I am :ou{aL Fitness: 3**

**Generation: 91 String: I am :ou{aL Fitness: 3**

**Generation: 92 String: I am :ou{aL Fitness: 3**

**Generation: 93 String: I am :ou{aL Fitness: 3**

**Generation: 94 String: I am :ou{aL Fitness: 3**

**Generation: 95 String: I am !ou{aL Fitness: 3**

**Generation: 96 String: I am !ou{aL Fitness: 3**

**Generation: 97 String: I am :ou{aL Fitness: 3**

**Generation: 98 String: I am :ou{aL Fitness: 3**

**Generation: 99 String: I am :ou{aL Fitness: 3**

**Generation: 100 String: I am :ou{aL Fitness: 3**

**Generation: 101 String: I am :ou{aL Fitness: 3**

**Generation: 102 String: I am :ou{aM Fitness: 3**

**Generation: 103 String: I am :ou{aL Fitness: 3**

**Generation: 104 String: I am :ou{aL Fitness: 3**

**Generation: 105 String: I am :ou{aL Fitness: 3**

**Generation: 106 String: I am :ou{aL Fitness: 3**

**Generation: 107 String: I am :ou{aL Fitness: 3**

**Generation: 108 String: I am :ou{aL Fitness: 3**

**Generation: 109 String: I am :ou{aL Fitness: 3**

**Generation: 110 String: I am :ou{aL Fitness: 3**

**Generation: 111 String: I am :ou{aL Fitness: 3**

**Generation: 112 String: I am oou{aL Fitness: 3**

**Generation: 113 String: I am :ou{ax Fitness: 3**

**Generation: 114 String: I am :ou{ax Fitness: 3**

**Generation: 115 String: I am :ou{aL Fitness: 3**

**Generation: 116 String: I am :ou{ax Fitness: 3**

**Generation: 117 String: I am :ou{aL Fitness: 3**

**Generation: 118 String: I am :ou{a[ Fitness: 3**

**Generation: 119 String: I am :ou=ax Fitness: 3**

**Generation: 120 String: I am :ou{aL Fitness: 3**

**Generation: 121 String: I am :ou{aL Fitness: 3**

**Generation: 122 String: I am :ou{aL Fitness: 3**

**Generation: 123 String: I am :ou{aL Fitness: 3**

**Generation: 124 String: I am :ou=ax Fitness: 3**

**Generation: 125 String: I am :ou{aL Fitness: 3**

**Generation: 126 String: I am :ou{aL Fitness: 3**

**Generation: 127 String: I am :oudaL Fitness: 3**

**Generation: 128 String: I am :ou{a] Fitness: 3**

**Generation: 129 String: I am :ou{aL Fitness: 3**

**Generation: 130 String: I am Fou{aL Fitness: 3**

**Generation: 131 String: I am :ou{aL Fitness: 3**

**Generation: 132 String: I am :ou{aL Fitness: 3**

**Generation: 133 String: I am :ou{aL Fitness: 3**

**Generation: 134 String: I am :ou{aL Fitness: 3**

**Generation: 135 String: I am :ou{aL Fitness: 3**

**Generation: 136 String: I am :ou{aL Fitness: 3**

**Generation: 137 String: I am ,ou{aL Fitness: 3**

**Generation: 138 String: I am Sou{a] Fitness: 2**

**Generation: 139 String: I am Sou{a] Fitness: 2**

**Generation: 140 String: I am Sou{a] Fitness: 2**

**Generation: 141 String: I am Sou{a] Fitness: 2**

**Generation: 142 String: I am Sou{a] Fitness: 2**

**Generation: 143 String: I am Sou{a] Fitness: 2**

**Generation: 144 String: I am Sou{a] Fitness: 2**

**Generation: 145 String: I am Sou{a] Fitness: 2**

**Generation: 146 String: I am Sou{a] Fitness: 2**

**Generation: 147 String: I am Sou{a] Fitness: 2**

**Generation: 148 String: I am Sou{a] Fitness: 2**

**Generation: 149 String: I am Sou{a] Fitness: 2**

**Generation: 150 String: I am Sou{a] Fitness: 2**

**Generation: 151 String: I am Sou{a] Fitness: 2**

**Generation: 152 String: I am Sou{a] Fitness: 2**

**Generation: 153 String: I am Sou{a] Fitness: 2**

**Generation: 154 String: I am Sou{a] Fitness: 2**

**Generation: 155 String: I am Sou{av Fitness: 1**

**Generation: 156 String: I am Sou{av Fitness: 1**

**Generation: 157 String: I am Sou{av Fitness: 1**

**Generation: 158 String: I am Sou{av Fitness: 1**

**Generation: 159 String: I am Sou{av Fitness: 1**

**Generation: 160 String: I am Sou]av Fitness: 1**

**Generation: 161 String: I am Sou{av Fitness: 1**

**Generation: 162 String: I am Sou]av Fitness: 1**

**Generation: 163 String: I am Sou]av Fitness: 1**

**Generation: 164 String: I am Sou]av Fitness: 1**

**Generation: 165 String: I am Sou]av Fitness: 1**

**Generation: 166 String: I am Sou]av Fitness: 1**

**Generation: 167 String: I am Sou{av Fitness: 1**

**Generation: 168 String: I am Sou{av Fitness: 1**

**Generation: 169 String: I am Sou]av Fitness: 1**

**Generation: 170 String: I am Sou]av Fitness: 1**

**Generation: 171 String: I am Sou{av Fitness: 1**

**Generation: 172 String: I am Sou{av Fitness: 1**

**Generation: 173 String: I am Sou]av Fitness: 1**

**Generation: 174 String: I am Sou$av Fitness: 1**

**Generation: 175 String: I am Sou{av Fitness: 1**

**Generation: 176 String: I am Sou[av Fitness: 1**

**Generation: 177 String: I am Sou{av Fitness: 1**

**Generation: 178 String: I am Sou{av Fitness: 1**

**Generation: 179 String: I am Sou{av Fitness: 1**

**Generation: 180 String: I am Sou]av Fitness: 1**

**Generation: 181 String: I am Sou[av Fitness: 1**

**Generation: 182 String: I am Sou]av Fitness: 1**

**Generation: 183 String: I am Sou[av Fitness: 1**

**Generation: 184 String: I am Sou[av Fitness: 1**

**Generation: 185 String: I am Sou]av Fitness: 1**

**Generation: 186 String: I am Sou[av Fitness: 1**

**Generation: 187 String: I am Sou]av Fitness: 1**

**Generation: 188 String: I am Sou{av Fitness: 1**

**Generation: 189 String: I am Sou{av Fitness: 1**

**Generation: 190 String: I am Sou]av Fitness: 1**

**Generation: 191 String: I am Sou]av Fitness: 1**

**Generation: 192 String: I am Soulav Fitness: 1**

**Generation: 193 String: I am Sou0av Fitness: 1**

**Generation: 194 String: I am Sou:av Fitness: 1**

**Generation: 195 String: I am Sou]av Fitness: 1**

**Generation: 196 String: I am Soulav Fitness: 1**

**Generation: 197 String: I am Sou]av Fitness: 1**

**Generation: 198 String: I am Sou{av Fitness: 1**

**Generation: 199 String: I am Sou{av Fitness: 1**

**Generation: 200 String: I am Sou]av Fitness: 1**

**Generation: 201 String: I am Soufav Fitness: 1**

**Generation: 202 String: I am Sou]av Fitness: 1**

**Generation: 203 String: I am Sou{av Fitness: 1**

**Generation: 204 String: I am Sou]av Fitness: 1**

**Generation: 205 String: I am Soulav Fitness: 1**

**Generation: 206 String: I am Sou]av Fitness: 1**

**Generation: 207 String: I am Soufav Fitness: 1**

**Generation: 208 String: I am Sou]av Fitness: 1**

**Generation: 209 String: I am Sou{av Fitness: 1**

**Generation: 210 String: I am Sou]av Fitness: 1**

**Generation: 211 String: I am Sou]av Fitness: 1**

**Generation: 212 String: I am Sou[av Fitness: 1**

**Generation: 213 String: I am Sou]av Fitness: 1**

**Generation: 214 String: I am Sou[av Fitness: 1**

**Generation: 215 String: I am Soulav Fitness: 1**

**Generation: 216 String: I am Soulav Fitness: 1**

**Generation: 217 String: I am Sou]av Fitness: 1**

**Generation: 218 String: I am Sou{av Fitness: 1**

**Generation: 219 String: I am Sou{av Fitness: 1**

**Generation: 220 String: I am Sou[av Fitness: 1**

**Generation: 221 String: I am Sou]av Fitness: 1**

**Generation: 222 String: I am Sou[av Fitness: 1**

**Generation: 223 String: I am Sou[av Fitness: 1**

**Generation: 224 String: I am Sou[av Fitness: 1**

**Generation: 225 String: I am Souyav Fitness: 1**

**Generation: 226 String: I am Soulav Fitness: 1**

**Generation: 227 String: I am Souyav Fitness: 1**

**Generation: 228 String: I am Soulav Fitness: 1**

**Generation: 229 String: I am Soumav Fitness: 1**

**Generation: 230 String: I am Souyav Fitness: 1**

**Generation: 231 String: I am Sou]av Fitness: 1**

**Generation: 232 String: I am Sou]av Fitness: 1**

**Generation: 233 String: I am Sou]av Fitness: 1**

**Generation: 234 String: I am Sou]av Fitness: 1**

**Generation: 235 String: I am Sou[av Fitness: 1**

**Generation: 236 String: I am Sou[av Fitness: 1**

**Generation: 237 String: I am Soulav Fitness: 1**

**Generation: 238 String: I am Sou]av Fitness: 1**

**Generation: 239 String: I am Soumav Fitness: 1**

**Generation: 240 String: I am Sou]av Fitness: 1**

**Generation: 241 String: I am Sou[av Fitness: 1**

**Generation: 242 String: I am Sou]av Fitness: 1**

**Generation: 243 String: I am SouAav Fitness: 1**

**Generation: 244 String: I am Sou]av Fitness: 1**

**Generation: 245 String: I am Sou]av Fitness: 1**

**Generation: 246 String: I am Sou]av Fitness: 1**

**Generation: 247 String: I am Sou[av Fitness: 1**

**Generation: 248 String: I am SouXav Fitness: 1**

**Generation: 249 String: I am Sou]av Fitness: 1**

**Generation: 250 String: I am Sou]av Fitness: 1**

**Generation: 251 String: I am Sou]av Fitness: 1**

**Generation: 252 String: I am SouRav Fitness: 1**

**Generation: 253 String: I am Sou]av Fitness: 1**

**Generation: 254 String: I am Sou]av Fitness: 1**

**Generation: 255 String: I am Sou]av Fitness: 1**

**Generation: 256 String: I am Sou]av Fitness: 1**

**Generation: 257 String: I am Sou]av Fitness: 1**

**Generation: 258 String: I am Sou]av Fitness: 1**

**Generation: 259 String: I am Sou]av Fitness: 1**

**Generation: 260 String: I am Sou]av Fitness: 1**

**Generation: 261 String: I am Sou]av Fitness: 1**

**Generation: 262 String: I am Sourav Fitness: 0**

**Process returned 0 (0x0) execution time : 0.457 s**

**Press any key to continue.**